## REMARKS

Claims 1-8 are pending in the Application.

Claims 1-8 stand rejected.

## I. OBJECTION TO THE DRAWINGS

The drawings have been objected to because FIGURES 5-8 are not clear. The Applicants have filed herewith formal drawings to replace the informal drawings filed on May 21, 1999.

## II. REJECTION UNDER 35 U.S.C. § 102

Claims 1-8 have been rejected under 35 U.S.C. § 102 as being anticipated by *Srinivasan, et al.*, U.S. Patent Application No. 2001/0051948A1 ("*Srinivasan*"). The Applicants respectfully traverse the rejection of claims 1-8 under 35 U.S.C. § 102.

Claim 1 is directed to a method for storing data that has at least some entries with multiple value attributes. The method includes the steps of profiling the data to determine whether the data should be stored in an attribute table, or, alternatively, in a merged table and an overflow table, and storing the data optimally based on the profiling step. The Examiner contends that *Srinivasan* discloses the method for storing data that has at least some entries with multiple attribute values including profiling the data to determine whether the data should be stored in an attribute table on the grounds that the attribute table of FIGURE 4 allegedly shows that the types, that is "EID", "ATTRNAME", "ATTRVAL" and "ATTRKIND" are sorted categories. (Paper No. 6, page 3.) The Examiner further asserts that this inherently indicates that the data must have been profiled before the table could be created. (Paper No. 6, page 3.) The Applicants respectfully disagree with the foregoing allegations for at least two reasons.

The table of FIGURE 4 is an attribute store table for entries in an exemplary directory information tree (DIT). (*Srinivasan*, page 3, ¶ 38.) Entries in the directory information tree are represented by one or more rows in the table. (*Id.*) The Examiner does not specifically state what "sorted categories" refers to. *Srinivasan* refers to additional system categories for object attributes that are stored in entries in the "ATTRKIND" column of the attribute store table (such as access and modification privileges). (*Srinivasan*, page 3, ¶ 40.) However, there is nothing therein that refers to sorting. The only reference to sorting the Applicants find in *Srinivasan* is with respect to catalog tables. (*Srinivasan*, page 5, ¶ 56.) *Srinivasan* teaches that catalog tables may be maintained in a sorted list of entries. *Id.*

Additionally, the Examiner does not provide any rationale explaining how the attribute table of FIGURE 4 inherently shows that the data must have been profiled before the table could be created. The Examiner must provide a basis in fact and/or technical reasoning to reasonably support the determination that the allegedly inherent characteristic necessarily flows from the teaching of the reference. MPEP § 2112. Indeed, there is nothing in the application of *Srinivasan* to the rejection of claim 1 that explains profiling as interpreted by the Examiner. Moreover, the allegedly inherent profiling in *Srinivasan* is not related to a determination whether the data should be stored in an attribute table, or alternatively in a merged table and an overflow table. The Examiner relies on FIGURE 5 of *Srinivasan* as disclosing a merged table and an overflow table. However, FIGURE 5 is an attribute store table similar to FIGURE 4, but including additional entries (that is, rows) that describe metadata associated with a particular entry (**100**) of the exemplary DIT. (*Srinivasan*, page 4, ¶44.) Nothing in *Srinivasan* has been shown to teach that the attribute store table of FIGURE 4 is merged table and the attribute store table of FIGURE 5 is an overflow table, or *vice versa*. Also, nothing identified in *Srinivasan* discloses that data is alternatively stored in an attribute table of FIGURE 4 or an attribute table of FIGURE 5, and, nothing identified in *Srinivasan* discloses that the data is profiled to determine whether it should be stored in an attribute table without subschema entries to define metadata (FIGURE 4) or including subschema entries to define metadata (FIGURE 5). (Metadata refers to information that describes the data in the system, such as information describing the structure and parameters of the tables and data maintained in the system. *Srinivasan*, page 3,

¶ 42.) The teaching referred to in Paper No. 6 discusses the content of the ATTRVAL for subschema entries (indicated by the value "2" in the EID column). (*Srinivasan*, page 4, ¶¶ 45, 46.) The teaching relied upon further discloses that the ATTRVAL column of a subschema entry can also identify the quantity of values to be provided for the defined attribute type. (*Srinivasan*, page 4, ¶47.) Likewise, these teachings do not discuss storing data optimally based on a profiling step. With respect to the optimal storing, the Examiner also refers to the teaching in *Srinivasan* that discusses storing data in a normalized format that is optimized for querying and searching. (Paper No. 6, page 3) (citing *Srinivasan*, page 7, ¶ 77). By the plain terms thereof, this disclosure refers to searching and querying optimization, not optimal storage of data.

Anticipation requires that a single prior art reference teach the identical invention as recited in the claim. MPEP § 2131. In other words, all of the limitations of the claim, arranged as required by the claim must be taught by the reference. *Id.* Because *Srinivasan* has not been shown to teach the identical invention of claim 1, the Applicants respectfully contend that claim 1 is allowable under 35 U.S.C. § 102 over *Srinivasan*.

Claim 2 depends from claim 1 and recites the method thereof in which the entries with single value attributes are stored in the merged table. The Examiner relies on the tables in FIGURES 2C and 5 which the Applicants understand to be alleged to teach the merged table as recited in claim 2. (*See* Paper 6, page 3.) Assuming, for the sake of argument, that either or both of the tables in FIGURES 2C and 5 of *Srinivasan* teach a merged table, there is nothing identified with respect thereto that discloses that the entries are single value attributes. Indeed, on the contrary, paragraph 47 of *Srinivasan* teaches that the attributes may have multiple values, and the Examiner has relied on pages 4-6, of *Srinivasan* as disclosing multiple value attributes. (*See* Paper No. 6, page 3) (rejecting claim 1 over *Srinivasan* on the ground that *Srinivasan* discloses a method for storing data that has at least some entries with multiple value attributes). In particular, *Srinivasan* teaches that a subschema entry could identify whether an attribute type comprises either single value, or multiple values of that attribute. (Note that whether the attribute is single-valued or multi-valued is defined by the schema, not whether a particular attribute is assigned multiple values or only a single value. (*Srinivasan*, page 4, ¶ 47.) For example, the country ('c') attribute defined by the LDAPv3

user schema to be single valued. *See* RFC 2256 p.3 (1997).) Thus, the Applicants respectfully contend that *Srinivasan* has not been shown to teach the identical invention of claim 2, and therefore, claim 2 is allowable under 35 U.S.C. § 102 over *Srinivasan*.

Claim 3 depends from claim 1 and is directed to the method thereof in which entries with multiple-value attributes are stored in the overflow table. The Examiner refers to the telephone number and manager catalog tables of FIGURES 6C and 6D, respectively. (Paper No. 6, page 4.) These catalog tables are maintained as indexes into the attribute store tables. (*Srinivasan*, page 5, ¶ 54.) For each attribute type that is indexed, a separate catalog table is maintained. (*Id.*) Each catalog table contains two columns, the first contains the EID of an entry or object having an attribute of the catalog attribute type, and the second provides the attribute value for the corresponding EID. (*Id.*) The Applicants note that each of the tables illustrated includes a single attribute value. The Examiner notes that *Srinivasan* states that a subschema entry could identify whether an attribute type comprises either a single value or multiple value attributes, however, nothing is disclosed in *Srinivasan* that the catalog tables illustrated are multiple-value attributes. Consequently, there is no justification for concluding that the catalog tables as taught by *Srinivasan* disclose overflow tables as recited in claim 3. (These tables are exemplary and a catalog table as taught by *Srinivasan* for the country attribute would be directed to a single value attribute in accordance with the LDAP schema.) Additionally, assuming, for the sake of argument, that the tables in FIGURES 5, 6C and 6D disclose overflow tables as recited in claim 3, then, at least with respect to the table in FIGURE 5, it cannot be a merged table as asserted with respect to claim 2. In other words, the table cannot be both a merged table and an overflow table. Thus, the Applicants respectfully assert that *Srinivasan* has not been shown to teach the identical invention of claim 3, and therefore claim 3 is allowable under 35 U.S.C. § 102 over *Srinivasan*.

Claim 4 is directed to the method of claim 1 in which the overflow table is an attribute table. The Examiner again refers to FIGURES 5, 6C and 6D as showing per attribute tables. As an initial matter, the Applicants respectfully disagree that FIGURE 5 of *Srinivasan* shows a per attribute table. Referring to FIGURE 5, FIGURE 5 shows table entries corresponding to at least eight attributes. With respect to the catalog tables of

FIGURES 6C and 6D, which are exemplary, *Srinivasan* makes no distinction between single-valued attributes and multi-valued attributes with respect to the entries in the catalog tables. Indeed, as the Examiner has noted, attributes may be either single-valued attributes or multi-valued attributes. However, there is nothing in the discussion of the catalog tables that indicates that the attributes corresponding to their respective tables are particularly one or the other, that is single-valued or multi-valued. Additionally, *Srinivasan* teaches that an attribute type can be modified by editing the appropriate subschema entry in the attribute table including, modifying a single-valued attribute type to be a multi-valued attribute type. (*Srinivasan*, page 8, ¶ 93.) Therefore, the Applicants respectfully contend that *Srinivasan* does not show an overflow table, as recited in claim 1, from which claim 4 depends which is an attribute table. Additionally, claim 4 incorporates the limitations of claim 1, and as previously discussed, *Srinivasan* has not been shown to teach the identical invention of claim 1, and therefore necessarily does not teach the identical invention of claim 4. For at least these reasons, the Applicants respectfully assert that claim 4 is not anticipated by *Srinivasan* and is thus allowable under 35 U.S.C. § 102 over *Srinivasan*.

Claim 5 is directed to the method of claim 1 in which a majority of the data is stored in a merged table and a set of additional values for the multiple-value attributes are stored in the overflow table. The Examiner asserts that FIGURES 2C and 5 exemplify a merge table in which a majority of single values are stored. (Paper No. 6, page 4.) The Applicants respectfully disagree. There is nothing that distinguishes the attributes in FIGURES 2C and 5 as either single-valued attributes or multi-valued attributes. In this respect, as the Applicants previously noted, in accordance with the LDAP schema, whether an attribute is single-valued or multi-valued is a "property" of the attribute. That is, whether a particular attribute admits, or may be assigned, multiple values or only single values, respectively is set by the schema. A single-valued attribute is not an attribute that in a particular embodiment has only a single value associated therewith and a multi-value attribute is not an attribute that in a particular embodiment has more than one value assigned to it. Thus, although the tables in FIGURES 2C and 5 illustrate only single values assigned to each of the exemplary attributes therein, it cannot be concluded that these are single-valued attributes. *Srinivasan* does not, for the purposes of the table entries in the tables of FIGURES 2C and 5, distinguish

between single-valued attributes and multi-valued attributes. The Examiner further asserts that the tables in FIGURES 6C and 6D are tables with multiple attributes for an instant entry of table 5. (Paper No. 6, page 4.) Again, the Applicants respectfully disagree. Referring to FIGURES 6C and 6D of *Srinivasan*, these figures incontrovertibly show a single attribute value for each entry. (*Srinivasan*, FIGURES 6C and 6D.) Note that each entry corresponds to a node in the DIT, which node is identified by the value in the EID entry in the respective tables. Thus, the Applicants also respectfully disagree with the Examiner's assertion that these tables illustrate more than one manager and/or telephone per person. Again, although the attribute types corresponding to each of the catalog tables may be either single-valued attributes or multi-valued attributes, *Srinivasan* does not distinguish between these for the purposes of the catalog tables of which FIGURES 6C and 6D are exemplary, nor has the Examiner identified teaching in *Srinivasan* to the contrary. For at least the foregoing reasons, the Applicants respectfully contend that *Srinivasan* does not teach the identical invention of claim 5. Consequently, claim 5 is allowable under 35 U.S.C. § 102 over *Srinivasan*.

Claim 6 is directed to the method of claim 1 in which the profiling step parses the data to identify entries with single-value attributes. The Examiner relies on the teaching in *Srinivasan* with respect to the ATTRVAL column of a subschema entry which can be used to identify the quantity of values to be provided for the defined attribute type, for example, a parameter that specifies a minimum or maximum number of telephone number values allowed for that attribute. (Paper No. 6, page 4) (citing *Srinivasan*, page 4, ¶ 47). This teaching does not refer to a step of parsing data. As *Srinivasan* teaches, subschema entries are rows that define metadata inserted in the attribute store table. (*Srinivasan*, pages 3-4, ¶ 43.) Metadata is information that describes data in the system and particular, that describes the structure and parameters of database and data maintained in the system. (*Srinivasan*, page 3, ¶ 42.) The Examiner also relies on the inherency discussed hereinabove in conjunction with claim 1. (Paper No. 6, page 4.) The Applicants have addressed the reliance on inherency hereinabove. For the reasons discussed in conjunction with claim 1, and the foregoing reasons with respect to the teachings in *Srinivasan*, paragraph 47, the Applicants

respectfully contend that *Srinivasan* has not been shown to teach the identical invention of claim 6. Consequently, claim 6 is allowable under 35 U.S.C. § 102 over *Srinivasan*.

Claim 7 is directed to the method of claim 1 in which the profiling step parses the data to identify given operations that are performed on the data once stored. Because, for the reasons discussed hereinabove in conjunction with claim 1, *Srinivasan* has not been shown to teach the profiling step as recited in claim 1, *Srinivasan* necessarily fails to teach the invention of claim 7. Therefore, claim 7 is allowable under 35 U.S.C. § 102 over *Srinivasan*.

Claim 8 depends from claim 1 and is directed to the method thereof in which the data is stored in a relational database backing store. Again, claim 8 incorporates all the limitations of claim 1 from which it depends, and which as discussed hereinabove is allowable under 35 U.S.C. § 102 over *Srinivasan*. Consequently, claim 8 is also allowable under 35 U.S.C. § 102 over *Srinivasan*.

## III. CONCLUSION

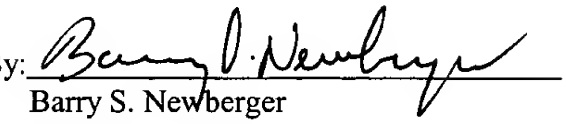
As a result of the foregoing, it is asserted by Applicants that the remaining claims in the Application are in condition for allowance, and respectfully request an early allowance of such claims.

Applicants respectfully request that the Examiner call Applicants' attorney at the below listed number if the Examiner believes that such a discussion would be helpful in resolving any remaining problems.

Respectfully submitted,

WINSTEAD SECHREST & MINICK P.C.

Attorney for Applicants

By: 

Barry S. Newberger
Reg. No. 41,527

5400 Renaissance Tower
1201 Elm Street
Dallas, Texas 75270-2199
(512) 370-2808

AUSTIN_1\225063\1
7047-P398US

Network Working Group
Request for Comments: 2256
Category: Standards Track

M. Wahl
Critical Angle Inc.
December 1997

# A Summary of the X.500(96) User Schema for use with LDAPv3

## 1. Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.



IESG Note

This document describes a directory access protocol that provides both read and update access. Update access requires secure authentication, but this document does not mandate implementation of any satisfactory authentication mechanisms.

In accordance with RFC 2026, section 4.4.1, this specification is being approved by IESG as a Proposed Standard despite this limitation, for the following reasons:

a. to encourage implementation and interoperability testing of these protocols (with or without update access) before they are deployed, and

b. to encourage deployment and use of these protocols in read-only applications. (e.g. applications where LDAPv3 is used as a query language for directories which are updated by some secure mechanism other than LDAP), and

c. to avoid delaying the advancement and deployment of other Internet standards-track protocols which require the ability to query, but not update, LDAPv3 directory servers.

Readers are hereby warned that until mandatory authentication mechanisms are standardized, clients and servers written according to this specification which make use of update functionality are UNLIKELY TO INTEROPERATE, or MAY INTEROPERATE ONLY IF AUTHENTICATION IS REDUCED TO AN UNACCEPTABLY WEAK LEVEL.



Implementors are hereby discouraged from deploying LDAPv3 clients or servers which implement the update functionality, until a Proposed Standard for mandatory authentication in LDAPv3 has been approved and published as an RFC.

## 2. Abstract

This document provides an overview of the attribute types and object classes defined by the ISO and ITU-T committees in the X.500

5.2. aliasedObjectName

The aliasedObjectName attribute is used by the directory service if the entry containing this attribute is an alias.

```
( 2.5.4.1 NAME 'aliasedObjectName' EQUALITY distinguishedNameMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 SINGLE-VALUE )
```

5.3. knowledgeInformation

This attribute is no longer used.

```
( 2.5.4.2 NAME 'knowledgeInformation' EQUALITY caseIgnoreMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{32768} )
```

5.4. cn

This is the X.500 commonName attribute, which contains a name of an object. If the object corresponds to a person, it is typically the person's full name.

```
( 2.5.4.3 NAME 'cn' SUP name )
```



5.5. sn

This is the X.500 surname attribute, which contains the family name of a person.

```
( 2.5.4.4 NAME 'sn' SUP name )
```

5.6. serialNumber

This attribute contains the serial number of a device.

```
( 2.5.4.5 NAME 'serialNumber' EQUALITY caseIgnoreMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.44{64} )
```

5.7. c

This attribute contains a two-letter ISO 3166 country code (countryName).

```
( 2.5.4.6 NAME 'c' SUP name SINGLE-VALUE )
```

5.8. l

This attribute contains the name of a locality, such as a city, county or other geographic region (localityName).

```
( 2.5.4.7 NAME 'l' SUP name )
```

5.9. st

This attribute contains the full name of a state or province (stateOrProvinceName).

```
( 2.5.4.8 NAME 'st' SUP name )
```

5.10. street

This attribute contains the physical address of the object to which